Xinyu AI Search: Enhanced Relevance and Comprehensive Results with Rich Answer Presentations [Scalable Data Science]

Bo Tang*
AIDS and SIAR, University of Science and Technology of China
MemTensor

Junyi Zhu*
ESAT-PSI, KU Leuven
Belgium

Hao Wang
Chenyang Xi, Yunhang Ge
MemTensor

Jiahao Wu
The Hong Kong Polytechnic University

Beihong Jin
Institute of Software, Chinese Academy of Sciences

Tingjian Ge
University of Massachusetts Lowell

Yuchen Feng, Yijun Niu
Wenqiang Wei, Yu Yu
Chunyu Li, Zehao Lin
MemTensor

Hao Wu, Ning Liao
Yebin Yang, Jiajia Wang
Zhiyu Li, Feiyu Xiong
MemTensor

Jingrun Chen†
jingrunchen@ustc.edu.cn
AIDS and SIAR, University of Science and Technology of China

Figure 1: Online evaluation (on June 1st, 2025) of Xinyu AI search engine for the query "Trump latest news". Xinyu features timeline visualization (right column), textual-visual choreography mechanism and sentence-level citation.

ABSTRACT

Generative AI search engines offer a key advantage over traditional search engines through their ability to synthesize fragmented information for queries, yet they still require improvements in relevance, comprehensiveness, and presentation. To these ends, we introduce Xinyu AI Search, a sophisticated system that incorporates a graph-based query decomposition to dynamically break down input queries into sub-queries, enabling more precise, stepwise retrieval and generation. Our retrieval pipeline enhances diversity through multi-source aggregation and query expansion, while filtering and re-ranking strategies optimize passage relevance. Additionally, Xinyu AI Search presents a specifically designed approach for fine-grained built-in citations and innovates in result presentation by integrating timeline visualization and textual-visual choreography. Evaluated on recent real-world queries, Xinyu AI Search outperforms eight existing systems in human (expert) assessments, excelling in relevance, comprehensiveness, and insightfulness. Our work, through the first public and comprehensive disclosure of this industry-leading full-stack commercial AI search engine, offers both a foundational reference for the community and, critically, demonstrates the effectiveness of the proposed system via rigorous evaluations and ablation studies, intended to catalyze further research and development in advanced AI search technologies.

PVLDB Reference Format:
Bo Tang, Junyi Zhu, Hao Wang, Chenyang Xi, Yunhang Ge, Jiahao Wu, Beihong Jin, Tingjian Ge, Yuchen Feng, Yijun Niu, Wenqiang Wei, Yu Yu, Chunyu Li, Zehao Lin, Hao Wu, Ning Liao, Yebin Yang, Jiajia Wang, Zhiyu Li, Feiyu Xiong, and Jingrun Chen. Xinyu AI Search: Enhanced Relevance and Comprehensive Results with Rich Answer Presentations [Scalable Data Science]. PVLDB, 14(1): XXX-XXX, 2020.
doi:XX.XX/XXX.XX

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

*Co-first author.
†Corresponding author.

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://anonymous.4open.science/r/AI-Search-33FB (including the Supplementary Material).

1 INTRODUCTION

In the era of information, a significant volume of events, knowledge, and resources has been digitized and made accessible through the Internet. As users continually strive to quickly and accurately locate relevant information within this rapidly growing digital ecosystem, the development of search engines has emerged as a critical solution to meet their fundamental need for efficient and reliable information retrieval [13, 48]. However, traditional search engines often face challenges in addressing ambiguous or complex queries. Furthermore, these systems typically present results as a ranked list, requiring users to manually synthesize information from diverse sources. This significantly increases comprehension efforts, particularly in scenarios where aggregating fragmented information from multiple resources is required.

Previously, the field of language modeling has been significantly advanced by the development of autoregressive models based on Transformer architectures [59, 70]. These architectures enable efficient parallel processing of sequences and facilitate large-scale unsupervised pretraining [21, 35, 59, 60], leading to the emergence of intelligent behaviors. Additionally, reinforcement learning from human feedback [18, 56] has further refined these models by aligning their outputs with human preferences and instructions. Together, these advancements have empowered machines to comprehend long-form text, interpret human intent, and generate responses that closely resemble human communication. Modern large language models (LLMs) have demonstrated human-level performance in tasks such as reading comprehension and reasoning within specific contexts [21, 43, 64, 78]. Their vast parameters also enable the encoding of extensive knowledge [14, 61]. Despite these strengths, LLMs continue to face critical challenges, including outdated knowledge, hallucinations and loss of attention [38, 42, 83]. These issues significantly undermine their reliability and limit their effectiveness in real-world applications.

More recently, retrieval-augmented generation (RAG) has emerged as a promising framework that integrates information retrieval techniques, such as search engines, with LLMs [38]. Studies have demonstrated that with externally retrieved non-parametric information, RAG can substantially reduce hallucinations in generated outputs while enabling LLMs to provide up-to-date information through in-context learning [10, 33, 53]. In turn, LLMs can enhance search quality by rewriting queries to better align with search engine requirements, improve readability of search results by synthesizing fragmented information from multiple retrieved sources, and summarize long-form texts [14, 21, 45, 79].

Building on the concept of RAG, generative AI search engines such as Perplexity AI [5], Tiangong AI [6], and Metaso [50] have emerged to provide synthesized answers using LLMs, rather than merely returning links like traditional search engines. While conversational LLM-based products like Gemini, DeepSeek and ChatGPT also employ RAG for up-to-date information and factual accuracy,

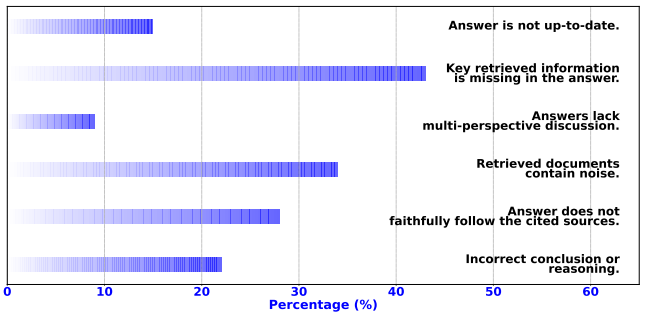

Figure 2: Common issues in generative AI search answers.

generative AI search engines distinguish themselves by emphasizing their dual role as both *search engines* and *reading tools*. This focus leads them to prioritize comprehensiveness and enhance the overall reading experience through advanced answer presentation.

A survey conducted in the United States found that over a quarter of adults considered switching to AI search engines in 2023 [63]. Despite the promising market expansion of AI search engines, many challenges remain in this field. Existing commercial AI search engines may produce inaccurate or unfaithful answers. We collected 100 queries across eight domains, and experts from our stakeholders identified several common issues in existing technologies. Fig. 2 presents the issues identified in Perplexity AI (based on GPT-4).

The issues outlined in Fig. 2 affect the relevance, comprehensiveness and overall quality of answers. Beyond these challenges, improving the reading experience represents another key aspect for enhancing the appeal of these novel search technologies. To tackle these challenges, we developed `Xinyu` (which indicates "a new way to present" in Chinese). `Xinyu` is a commercialized (B2B) product and developed with a focus not only on achieving competitive performance against existing technologies and incorporating novel features, but also on ensuring deployment stability and safety. In this paper, we provide a detailed breakdown of Xinyu AI search. An online test showcase is presented in Fig. 1. **Our contributions can be summarized as follows:**

- We propose Xinyu AI Search, a generative AI search engine specifically designed to address critical challenges including irrelevance, incompleteness, and fragmented presentation in existing search systems. We provide a detailed industrial-level architecture of Xinyu, illustrating the entire retrieval and generation workflow, from query processing to answer presentation. To the best of our knowledge, this is the first comprehensive public disclosure of a full-stack design for a commercialized generative AI search engine.
- We present a novel Graph-based Query Decomposition (GQD) method to systematically handle complex queries by decomposing them into structured sub-queries. By integrating this approach with multi-source retrieval (enhancing recall comprehensiveness) and fine-tuned reranking techniques (filtering irrelevant passages), Xinyu significantly improves the system's capability to address complex information needs. Experimental results show that our approach notably improves comprehensiveness by at least 0.891 points out of a 10-point scale, compared to eight competitive baseline systems.

- We improve the user reading experience with advanced presentation techniques, including fine-grained citations, timeline visualization, and textual-visual choreography. These techniques greatly enhance the clarity, verifiability, and readability of search results, achieving the highest insightfulness score among all evaluated systems, surpassing others by at least 0.537 points on a 10-point scale.
- Comprehensive evaluations involving over 300 real-world queries and more than 81,000 expert assessments confirm Xinyu's superior performance. Specifically, Xinyu achieves the highest average evaluation score of 9.235, compared to the baseline average of 8.810, confirming its effectiveness and advantages across multiple critical dimensions.

2 RELATED WORK

Constructing a generative AI search engine involves the design of three main components: retrieval, contextual generation and orchestration [26]. Since these components relate to a broad range of research topics, we introduce works closely related to this paper.

2.1 Retrieval

Effective retrieval is critical to system performance, as the quality of retrieved information significantly shapes the final output. Query rewriting techniques aim to transform user queries into more precise and retrieval-friendly formats, addressing ambiguities and enhancing alignment with indexed data [24, 46, 57, 84]. Similarly, query expansion enriches the input by generating alternative or supplementary queries, ensuring the retrieval of a broader and more contextually relevant set of documents [22, 30, 71]. The choice of retrieval sources also impacts system performance, with strategies leveraging unstructured data, semi-structured data, and structured knowledge graphs to provide domain-specific and fine-grained knowledge [27, 44, 72, 81]. Lastly, when database construction is needed, effective chunking strategies, metadata enrichment, and hierarchical indexing are considered to ensure that retrieval components operate efficiently [29, 51, 68].

2.2 Contextual Generation

After retrieving documents for a query, the generation process relies on their context to produce accurate and well-informed responses. However, this process is susceptible to two key challenges. First, studies show that irrelevant information in the references can distract the model and lead to inaccurate answers [19, 80]. Moreover, various techniques have been developed to select or focus on the most salient information. These strategies can be broadly categorized into two main pathways: manipulating the LLM's input context and optimizing the LLM itself. The first pathway focuses on refining the model's input. This includes techniques such as **reference filtering**, which aims to eliminate irrelevant or noisy retrieved documents entirely [20, 47]; **context selection**, which identifies the most relevant portions of the retrieved context while discarding less pertinent parts [32, 77]; and **reference reranking**, which reorganizes information to position the most critical content more prominently [25, 88]. The second pathway involves adapting the model itself. Some work employs fine-tuning to adapt a language model for specific tasks or domains, or to align its outputs with desired formats and styles, thereby improving task performance [23, 40, 87].

2.3 Orchestration

Simple pipelines that directly generate responses from retrieved results often fall short, motivating the research into auxiliary components and the orchestration of more sophisticated workflows. Iterative workflows involve alternating between retrieval and generation processes, progressively enriching the context by utilizing generated text or intermediate results to refine subsequent retrievals [62]. Adaptive workflows enhance system flexibility by dynamically determining the necessity of retrieval based on the context of the query, often incorporating mechanisms for self-assessment and adjustment [9, 33, 52]. Recursive workflows break down complex queries into smaller, interdependent subtasks, iteratively resolving each to produce comprehensive and logically structured responses [36, 69]. The chain-of-knowledge strategy first generates rationales for answering a query, then leverages retrieval results to refine these rationales and deduce the final response [41]. Recent work uses reinforcement learning to determine when to generate retrieval queries during the orchestration, thereby improving the accuracy of open-ended question answering [34, 85].

Compared to existing work, our paper presents a complete design of a generative AI search engine instead of multiple standalone algorithms, aiming to provide users with comprehensive and visually engaging answers in response to their queries.

3 XINYU AI SEARCH

In this section, we first give an overview of the Xinyu AI search, and then elaborate on each component in Xinyu AI search in detail.

3.1 Overview

The Xinyu AI search, as shown in Fig. 3, is a pipeline comprising query preprocessing and decomposition, multi-step retrieval, contextual generation, and visual presentation.

The Xinyu AI search can integrate a pre-trained LLM, a reranker and a text embedding model, and fine-tune first two models using data from existing public datasets, as well as synthetic (or labeled) data generated by stronger models (e.g., larger models), after these data are refined through expert selection [8].

Specifically, let $\mathcal{D}$ denote the training dataset. For fine-tuning generative LLM π_θ, we adopt pairs of user query $x \in D$ and ground-truth answer y as training data, and the following next-token prediction (NTP) loss as the optimization objective:

$$\mathcal{L}_{\text{NTP}}(\theta) = -\mathbb{E}_{(x,y)}\left[\sum_{t=1}^{|y|} \log \pi_\theta(y_t|x, y_{<t})\right], \tag{1}$$

where θ denotes the model parameters and $y_{<t}$ represents preceding tokens. For fine-tuning the reranker, let $h_\phi(\cdot)$ denote the score function implemented by a small LLM [17] with parameters ϕ. We optimize the LLM to correctly predict the positive sample among the negatives using an InfoNCE loss [54] over the scored samples:

$$\mathcal{L}_{\text{Re}}(\phi) = -\mathbb{E}_{(x,x^+,x^-_{1:N})}\left[\log \frac{e^{h_\phi(x,x^+)}}{e^{h_\phi(x,x^+)} + \sum_{i=1}^{N} e^{h_\phi(x,x^-_i)}}\right]. \tag{2}$$

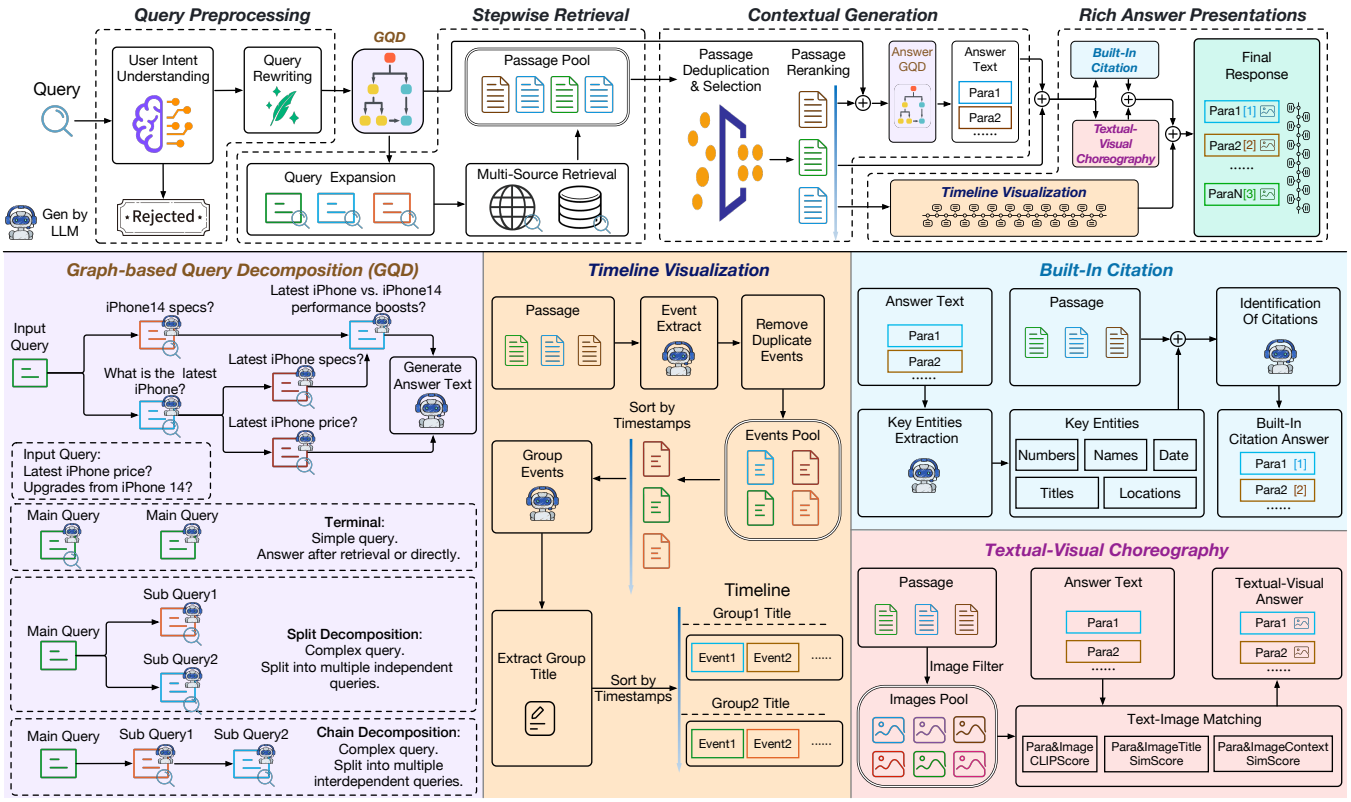

Figure 3: Xinyu AI search. The upper row illustrates the full response pipeline, while the lower row provides a more detailed depiction of several novel approaches integrated into this AI search engine.

where x, x^+ and x^- denote an anchor sample, a positive sample and a negative sample, respectively.

3.2 Query Preprocessing

Upon receiving a query, a fine-tuned LLM (currently based on Qwen-2.5-14B) is employed to filter out unsafe or harmful queries and clarifies ambiguous ones by prompting the user with options, such as suggesting a specific region for a general query like "the current state of the economy." If the query is ambiguous or lacks specificity, the LLM prompts the user with clarifying questions or options to refine his/her intent. The query might be rejected if it is regarded as one that warrants refusal. See Supplementary Materials D.1 for the detailed description of this fine-tuning process for this task.

Following this, any open-source LLM (currently Qwen-2.5-14B) is used to rewrite the query by translating relative spatiotemporal terms (e.g., “last week”) into precise timestamp ranges, resolving vague locations (“nearby”) to specific places, and canonicalizing incomplete entity names.

3.3 Query Decomposition

To overcome the limitations of naive RAG in capturing nuanced and multi-faceted information for non-elementary queries, we propose a novel and practical decomposition strategy that breaks down the user’s query into sub-queries and answers them stepwise. Specifically, we use a graph structure to represent the decomposed sub-queries due to its flexibility in representing sequential and lateral dependencies (*see Fig. 3 for a concrete example*). This graph structure distinguishes Xinyu from existing technologies using linear and tree decomposition [82, 86]. Specifically, the graph structure precisely captures parallel and joint dependencies among sub-queries, enabling finer-grained decomposition. We refer to this approach as the graph-based query decomposition (GQD).

As shown in Fig. 3, after query rewriting, a single query is transformed into a GQD. In the GQD, nodes represent sub-queries, while directed edges indicate dependencies. Given a query, we use a fine-tuned generative LLM (Qwen2.5-72B) to construct the corresponding GQD by defining nodes and their pairwise relationships. The LLM is instructed with the GQD definition and few-shot examples. Guided by the prompt, the LLM applies the following decomposition strategies (often a combination of them to fully break down an input query): **1)** *Chain decomposition:* A query is sequentially decomposed into a series of sub-queries, where each parent node provides preliminary information for its child node and the descendant nodes, e.g. least-to-most [86]. **2)** *Split decomposition:* The query is divided into multiple independent sub-queries. **3)** *Terminal:* The input query is elementary, and decomposition is unnecessary. For fine-tuning, we collect a set of queries and instruct multiple models using the prompt provided in Supplementary Material B.1

to generate GQDs. The generated GQDs are then programmatically validated to ensure that the parent-child relationships meet the specified requirements, and duplicate GQDs are removed. Finally, human experts examine the data and select correctly generated high-quality samples. In the end, the left samples are used for fine-tuning, adopting the loss function in Eq. (1).

GQD also determines the workflow for subsequent generation operations: the parent node is processed first, while the child node generates its output based on retrieved documents, ancestor sub-queries, and their corresponding answers. With GQD as its core, `Xinyu` facilitates hierarchical reasoning and evidence aggregation, ensuring a logically consistent and comprehensive resolution of input queries.

3.4 Stepwise Retrieval

After constructing the GQD, we aggregate all sub-queries and perform retrieval. To ensure the retrieved documents capture diverse details necessary for generating answers, we enhance retrieval diversity through query expansion and multi-source retrieval, as discussed below. *For an illustration, refer to the top row of Fig. 3.*

3.4.1 Query Expansion. Given a sub-query, we ask an LLM to generate multiple relevant queries. These queries are called retrieval queries since they are subsequently entered into a search engine to retrieve more informantion. Specifically, the LLM is instructed to act as a subject matter expert in an university, expanding the given query to create related questions that assess students' comprehensive understanding of the topic across multiple dimensions: 1) content mastery, 2) understanding of key elements, 3) contextual analysis, and 4) extended thinking.

3.4.2 Multi-source retrieval. To ensure the recency and relevance of retrieved information, we submit each retrieval query to multiple search engines simultaneously, leveraging the ranking algorithms in search engines to obtain more comprehensive content and mitigate biases. Additionally, we incorporate `Milvus` and `Elasticsearch` for our B2B clients to build local retrieval systems.

Directly feeding raw retrieved documents (e.g., a web page) into an LLM can lead to high perplexity and degraded response quality. To make the input content LLM-friendly and safe, we implement a robust content filtering pipeline. This process involves removing disruptive elements, filtering sensitive or extraneous information, and standardizing formatting. More details on the filtering rules are provided in Supplementary Material C.

After filtering, we segment documents into passages using the `RecursiveCharacterTextSplitter` method [15]. We adopt a small chunk size (i.e., 350) with a relatively large overlap (i.e., 25%) to optimize the performance of the subsequent text embedding model, following the research findings by Azure AI Search [12].

3.5 Contextual Generation

Note that the retrieved passages corresponding to a sub-query vary in relevance and often contain duplicate information. Moreover, when generating a response for each sub-query, it is essential to adhere to the dependency structure in GQD, ensuring that parent nodes are processed before their child nodes. Further, directly employing any LLM to generate an answer for a query based on responses of sub-queries is not ideal. This is because LLMs tend to allocate different levels of attention to different sections of the input. To mitigate these problems, we employ a text embedding model and fine-tune a re-ranking model and a generative LLM to implement passage deduplication, re-ranking the passages and generate the answer for a query, *illustrated in the top row of Fig. 3.*

For passage deduplication, we use a fine-tuned text embedding model (bge-large [75]) to compute embeddings for the passages and then calculate pairwise cosine similarities. We aim to identify the largest subset of passages where the similarity score between any two passages is no greater than 0.8. Finding the optimal solution to this problem corresponds to solving the maximum independent set problem [67], which is NP-hard. To improve computational efficiency, we adopt a greedy strategy that processes each passage sequentially, retaining it only if its similarity to all previously retained passages remains below 0.8. For re-ranking, we finetune a reranker model (bge-reranker-v2-m3 [17, 39] to sort the passages based on their similarity to the sub-query. Details of the reranker model are provided in Supplementary Material D.2. For answer generation, a fine-tuned generative LLM (Qwen 2.5-72B) answers the sub-queries either sequentially or in parallel based on the dependency in the GQD. For nodes with ancestors, the cumulative Q&A of the ancestor chain is prepended to the candidate passages. When all leaf nodes are processed, their Q&A pairs are concatenated with the main query to form the final answer.

3.6 Rich Answer Presentations

Traditional chatbots often rely on linear text stacking, which can impose a high cognitive load on users. Given that AI-powered search engines facilitate extensive knowledge transmission, integrating cognitive scaffolding is essential to support user comprehension. Cognitive science research has shown that structured information and multimodal presentations enhance the efficiency of information assimilation [49, 58, 66]. Moreover, since mitigating hallucinations in LLMs remains challenging [83], aiding users in result verification and fostering confidence in synthesized outputs are crucial. To address these challenges, we incorporate timeline visualizations, textual-visual choreography, and built-in citations, as discussed below, to optimize the reading experience.

3.6.1 Built-In Citation. A straightforward approach to citation generation involves instructing the LLM to produce citations on the fly, as implemented by Lepton AI [3]. However, our initial evaluation indicates that this method exhibits a high error rate. Furthermore, we observe that some existing systems, such as Perplexity AI, place citations at the end of a paragraph, potentially detaching references from the corresponding evidence. To enhance both citation accuracy and granularity, we propose a novel citation scheme.

As illustrated in the bottom right of Fig. 3, `Xinyu` decouples answer generation from citation attachment. Our pipeline fine-tune two models. The first model, an SLM (Qwen2.5-3B), extracts key entities (e.g., dates, locations, names) from the generated answer on a *sentence-by-sentence basis*. If a sentence contains extractable entities, a second SLM (Qwen2.5-3B) identifies citations based on these entities, its orignal sentence, and retrieved documents. Both models have been fine-tuned for their respective tasks. Prompt details are provided in Supplementary Material B.4. In cases where no

Table 1: Multi-faceted comparison of different systems. Higher value indicates better performance, 10 is the maximum.

Model	Conciseness	Numerical Precision	Relevance	Factuality	Timeliness	Comprehensiveness	Clarity	Coherence	Insightfulness	Average
Perplexity AI [5]	**9.851**	9.630	9.436	8.524	8.553	7.284	9.612	9.853	6.543	8.810
Tiangong AI [6]	9.840	**9.722**	7.812	8.924	8.103	8.020	9.604	9.802	6.535	8.707
Ernie Bot [2]	9.770	9.320	8.883	8.028	8.406	7.798	9.524	**9.900**	5.963	8.621
KIMI [4]	9.840	9.515	8.529	8.224	8.966	8.155	9.223	9.709	6.796	8.773
Metaso [50]	9.760	8.941	8.515	7.408	8.403	5.689	9.383	9.689	4.759	8.061
ChatGLM [16]	9.810	9.420	8.949	**9.124**	8.346	6.168	9.533	9.726	5.047	8.458
Baichuan [1]	9.660	9.596	6.486	7.612	8.220	8.252	9.223	9.612	6.117	8.309
Tongyi [7]	9.803	9.009	7.586	7.212	8.194	7.677	9.293	9.899	5.859	8.281
Xinyu (Ours)	9.813	9.714	**9.533**	8.932	**9.205**	**9.143**	**9.633**	9.810	**7.333**	**9.235**

Table 2: Pearson correlation coefficients between human and LLM scores under different evaluation criteria.

Metric	Value	Metric	Value
Comprehensiveness	0.679	Conciseness	0.787
Numerical Precision	0.741	Clarity	0.737
Relevance	0.807	Coherence	0.746
Factuality	0.831	Insightfulness	0.610
Timeliness	0.759		

entities are extracted from a sentence, we adopt a fallback method that computes the sentence embedding using bge-large and assigns a citation if its cosine similarity with a retrieved document exceeds 0.6. To reduce latency, we implement an asynchronous processing strategy that runs citation assignment in parallel with answer generation (albeit with a one-sentence delay).

3.6.2 Timeline Visualization. In online search scenarios focused on news and events, integrating timeline visualizations enables users to better understand the evolution and context of events. We propose a novel timeline visualization scheme as illustrated in the bottom middle of Fig. 3. First, we collect all retrieved passages following the passage selection (see Sec. 3.5). Next, we instruct an LLM (Qwen2.5-14B) to extract any event time mentioned in each passage and to generate a corresponding title and summary. If a passage does not explicitly mention a time, we resort to using the document's report time as extracted by the same LLM. Passages lacking temporal information in both the passage and the document are discarded. Because the retained passages may describe the same content, we then employ `bge-large` to compute text embedding of the concatenated title and summary and calculate pairwise cosine similarities across passages. Passages with a similarity score exceeding 0.9 are merged by discarding the one with the later timestamp, resulting in a list of distinct events with timestamps. To make timeline visualization more structured, we further instruct the LLM to group these events and derive relevant keywords based on their summaries. Finally, we present the event titles for each group, sorted according to their timestamps. It is noteworthy that our timeline visualization scheme is integrated within the framework, where upstream pipeline components (GQD, retrieval, and passage ranking) enhance the relevance and precision of the displayed events. This integration distinguishes our solution from other methods [28, 74]. GQD decomposes complex queries into simpler sub-queries, enabling targeted retrieval, while passage ranking removes noise, together yielding more precise and context-aware results.

3.6.3 Textual-Visual Choreography. A picture is worth a thousand words. As illustrated in the bottom right of Fig. 3, `Xinyu` integrates relevant images into textual responses to enhance information assimilation. These images are extracted from retrieved documents. To ensure quality and relevance, we first filter out noisy images, retaining only those of high quality. Specifically, a rule-based filtering algorithm eliminates logos, icons, and low-resolution images. Subsequently, we compute the similarity between the textual description associated with the image and the main query using bge-reranker-v2-m3 and remove those smaller than 0.3. To determine the optimal placement of images, we compute the pairwise similarity between generated answer paragraphs and candidate images. This computation involves a weighted average of three measures: **(1)** the embedding cosine similarity between the generated text paragraph and the image, computed using the clip-vit-huge-patch14; **(2)** the estimated similarity between the synthesized paragraph and the retrieved document's title, obtained via bge-reranker-v2-m3; and **(3)** the embedding cosine similarity between the synthesized paragraph and the retrieved document's text using bge-large. Pairwise similarities are assembled into a matrix. Then we determine the optimal image-to-text alignment using the Hungarian algorithm [37].

4 EXPERIMENTS

4.1 Experimental Setup

Multi-faceted Evaluation Criteria. Due to the open-ended nature of answers in real scenarios, we adopted rating criteria for evaluating generated answers rather than computing their match to a fixed reference answer. We invited experts with journalism backgrounds and master's degrees to develop a multi-faceted rating criteria, including: **(1)** Conciseness, **(2)** Numerical Precision, **(3)** Relevance, **(4)** Factuality, **(5)** Timeliness, **(6)** Comprehensiveness, **(7)** Clarity, **(8)** Coherence, and **(9)** Insightfulness. More detailed definitions for each dimension are provided in Supplementary Material A. Our nine evaluation metrics were chosen based on two principles: **(1)** each metric is concise, easily interpretable, and independent; and **(2)** collectively, they capture the majority of observed issues.

Test Set. We sourced a set of over 300 recent queries from the news aggregator `TopHub` to serve as our test set. The queries in the test set were nearly uniformly distributed across eight distinct domains (i.e., Politics, Economy, Society, Technology, Sports, Entertainment, Military and History) with "Society" being the most prominent domain (18.00%) and "Sports" constituting the smallest share (7.33%). We find that over 300 high-quality, real-world queries

Table 3: Comparison of timeline visualization. Except wall time, higher value is better.

Approach	Timeliness	Comprehensiveness	Clarity	Event Count	Precision	Wall Time (s) ↓
CHRONOS [74]	7.02	5.79	6.00	5.12	79%	67.44
Xinyu (Ours)	**8.07**	**8.08**	**8.24**	**10.14**	**84%**	**33.27**

Table 4: Comparison of textual-visual choreography.

Approach	Inclusion (%) ↑	Precision (%) ↑
Metaso [50]	3.0	72.2
Xinyu (Ours)	**80.0**	**90.0**

Table 5: Comparison of response latency (time to first token, seconds).

Perplexity AI	Tiangong AI	ErnieBot	KIMI	Metaso	ChatGLM	Baichuan	Tongyi	Xinyu
13.9	14.2	4.0	6.0	2.8	10.4	7.9	6.1	10.4

Table 6: Comparison of built-in citation.

Model	Density (%) ↑	Precision (%) ↑
Perplexity AI [5]	46.6	82.1
Metaso [50]	59.5	49.7
Tiangong [6]	27.0	90.8
Baichuan [1]	45.7	**90.9**
KIMI [4]	41.4	72.9
Xinyu (Ours)	**67.2**	90.4

are sufficient to demonstrate the performance of different methods. It is also noteworthy that we required three ratings for each combination of system, query, and rating criterion. Just Tab. 1 alone contains over 81,000 human ratings, underscoring the sheer scale of our evaluation. The numerical evaluation results are mainly based on Chinese queries, as many of our expert evaluators are native Chinese speakers.

LLM Evaluation. Evaluating the generated answers for all experiments using multi-faceted criteria by human experts is prohibitively expensive. Therefore, we considered to employ an LLM to evaluate the generated text for a subset of experiments. To validate the effectiveness of the LLM as an evaluator, we also conducted a comparison experiment involving both human and LLM (GPT-4o-2024-08-06) evaluators and calculated the Pearson correlation between their final scores. As shown in Tab. 2, human and LLM scores exhibit a high correlation coefficient, indicating that it is feasible to use LLM as an evaluator. Hence, experiments reported in Tabs. 7 to 9, were evaluated with GPT-4o.

Baselines. We compare Xinyu with eight existing systems, including: **Generative AI search engines**: Perplexity AI [5], Metaso [50], Tiangong AI [6], ChatGLM [16], and Tongyi [7]; and **Conversational LLMs with RAG**: KIMI [4], Ernie Bot [2], and Baichuan [1]. For Perplexity AI, we selected GPT-4o [55] as the backend model.

4.2 Comparison with Existing Systems

4.2.1 Multi-Faceted Evaluation of the Generated Answer. We first compare Xinyu with existing systems across the multi-faceted evaluation criteria, using scores provided by human experts in a model-agnostic setting. As shown in Tab. 1, Xinyu performs competitively while achieving the highest average score (**9.235** vs. 8.810). Notably, Xinyu significantly outperforms other systems in comprehensiveness (**9.143** vs. 8.252, $\mathbf{p < 0.001}$ in t-tests) and insightfulness (**7.333** vs. 6.796, $\mathbf{p < 0.001}$ in t-tests). Each answer to a query was scored by at least three experts. To assess inter-rater consistency, we computed pairwise Pearson correlation coefficients for scores of experts. The average correlation across all pairs was **0.74**, indicating a high level of agreement.

4.2.2 Representation Enhancement.

Built-In Citation. We evaluate our approach using two key metrics. The first, citation precision, measures whether the provided evidence is genuinely supported by the cited source. The second, citation density, quantifies the proportion of sentences containing citations relative to the total number of sentences. Citation density reflects two factors: **(1)** the extent to which the generated answer relies on retrieved information and **(2)** the placement of citations. Some existing systems, such as Perplexity AI, often position citations at the end of a paragraph, making it difficult for users to trace specific claims, especially when multiple citations correspond to different parts of a paragraph stack. In such cases, citation density is also lower. As shown in Tab. 6, Xinyu achieves significantly higher citation density (**67.2** vs. 59.5) while maintaining competitive citation precision.

Timeline Visualization. A parallel study introduces CHRONOS for timeline generation [74]. We compare Xinyu against CHRONOS using three multi-faceted evaluation criteria—timeliness, comprehensiveness, and clarity—assessed by human evaluators. Additionally, we evaluate event count to measure the system's ability to identify multiple events, precision to assess the relevance of extracted events to the query, and wall time of online deployments to gauge computational efficiency. Tab. 3 demonstrates that Xinyu significantly outperforms CHRONOS across multiple dimensions.

Textual-Visual Choreography. Among the baselines, Metaso [50] also implements textual-visual choreography. We compare our method against it by evaluating two metrics: inclusion (the rate at which images are incorporated into the generated answers) and precision (the percentage of included images that are contextually relevant). As shown in Tab. 4, Xinyu outperforms Metaso significantly on both metrics.

4.2.3 Test-Time Latency. To improve test-time latency, we parallelize modules that can run concurrently, such as parallel sub-queries in the GQD and built-in citation generation. Additionally, we fine-tune and quantize our models, selecting the fastest one that meets our needs. As shown in Tab. 5, the response latency of Xinyu is comparable to existing technologies. The time for Xinyu is measured based on our deployment on a cluster of 16 Muxi MXC500 GPUs (each GPU has an equivalent computing power of 70% of an Nvidia A800). Baselines are evaluated based on their publicly available interfaces.

Table 7: Ablation study of sub-modules in our approach, "−" indicates skipping the sub-module.

Variant	Conciseness	Numerical Precision	Relevance	Factuality	Timeliness	Comprehensiveness	Clarity	Coherence	Insightfulness	Average
Full Approach	9.880	9.547	9.547	9.731	8.300	8.533	9.900	9.747	7.107	9.143
− Query Preprocessing	9.810	9.422	9.497	9.646	8.279	8.423	9.891	9.637	6.993	9.066
− Query Expansion	9.793	9.300	9.593	9.626	8.300	8.493	9.867	9.780	6.827	9.064
− GQD	9.780	9.607	9.413	9.731	8.320	8.620	9.860	9.827	6.993	9.127
− Passage Selection	9.833	9.473	9.513	9.717	8.207	8.613	9.847	9.787	7.060	9.118
− Passage Rerank	9.827	9.587	9.587	9.731	8.220	8.587	9.873	9.800	6.987	9.132

Table 8: Ablation study of replacing our fine-tuned LLMs with proprietary models.

Model	Conciseness	Numerical Precision	Relevance	Factuality	Timeliness	Comprehensiveness	Clarity	Coherence	Insightfulness	Average
GPT-4o [55]	9.828	9.425	**9.621**	9.433	7.973	8.473	9.753	9.717	6.520	8.972
Qwen 2.5-72B [76]	9.780	9.290	9.463	8.987	8.053	8.140	9.893	9.633	6.687	8.881
Xinyu (Ours)	**9.880**	**9.547**	9.547	**9.731**	**8.300**	**8.533**	**9.900**	**9.747**	**7.107**	**9.142**

Table 9: Ablation study of replacing our fine-tuned answer generation model with proprietary models.

Model	Conciseness	Numerical Precision	Relevance	Factuality	Timeliness	Comprehensiveness	Clarity	Coherence	Insightfulness	Average
GPT-4o [55]	**9.854**	9.482	**9.588**	9.597	8.107	8.515	9.849	9.734	6.989	9.080
Qwen 2.5-72B [76]	9.824	9.380	9.551	9.337	7.947	8.417	9.848	9.683	6.884	8.986
Xinyu (ours)	9.880	**9.547**	9.547	**9.731**	**8.300**	**8.533**	**9.900**	**9.747**	**7.107**	**9.142**

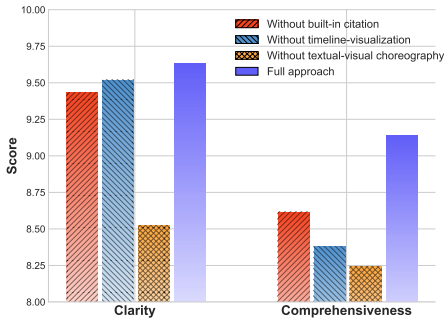

Figure 4: Ablation study by removing sub-modules in rich answer representation.

4.3 Ablation Study

Query and Retrieved Documents Processing. We first conduct an ablation study on the sub-modules designed to enhance text quality. Specifically, we compare the performance of the generated answers after omitting each sub-module against the full approach. LLM evaluation is used to rate the responses based on the multi-faceted evaluation criteria, with results provided in Tab. 7. Notably, skipping a sub-module does not always lead to a decline in all metrics. For example, omitting query expansion may improve relevance. However, the full approach, which integrates all sub-modules, achieves the best overall performance, as indicated by the average scores.

Representation Enhancement. We further conduct an ablation study on built-in citation, timeline visualization, and textual-visual choreography to assess their impact on clarity and comprehensiveness based on human evaluation. As shown in Fig. 4, removing any of these modules reduces the comprehensiveness of the generated answer. Additionally, textual-visual choreography has a strongly positive effect on clarity. These findings highlight the advantages of rich answer representations in supporting cognitive scaffolding and enhancing information assimilation efficiency.

Table 10: Ablation study of fine-tuning the rerank model.

Model	Precision	Recall	F1 Score	Wall Time (s)
GPT-4o [55]	0.717	0.719	0.692	3.6
Qwen 2.5-72B [76]	0.541	0.894	0.641	2.4
bge-reranker-v2-m3 [17]	0.568	0.671	0.562	0.1
Xinyu (ours)	0.607	0.735	0.623	0.1

Fine-Tuning. In `Xinyu`, we fine-tune two large language models for entity extraction, GQD and answer generation, and validate their effectiveness through three ablation studies. As shown in Tab. 8, replacing these modules with GPT-4o and Qwen-2.5-72B confirms that improvements in question rewriting, entity extraction, and GQD generation positively impact final answer quality. Tab. 9 demonstrates that fine-tuning the answer generation model yields consistent performance gains across eight evaluation metrics, including Conciseness and Numerical Precision. Tab. 10 further shows that our reranking model significantly outperforms the baseline while maintaining the inference latency of the base model, achieving performance comparable to large-scale language models.

5 CONCLUSION

In this paper, we present `Xinyu`, a generative AI search engine designed to tackle multi-faceted challenges in answer generation and improve the reading experience through a fully integrated pipeline. Our work not only builds on state-of-the-art research but also gives novel solutions to specific challenges. Extensive experimental results demonstrate the superiority of `Xinyu` over existing systems. Future work will focus on enhancing multilingual capabilities and expanding domain-specific optimizations.

REFERENCES

[1] Baichuan AI. 2024. Baichuan. https://ying.baichuan-ai.com/ Accessed: 2025-02-05.

[2] Baidu AI. 2024. Yiyan. https://yiyan.baidu.com/ Accessed: 2025-02-05.

[3] Lepton AI. 2025. Search with Lepton. https://github.com/leptonai/search_with_lepton Accessed: 2025-02-02.

[4] Moonshot AI. 2024. KIMI. https://kimi.moonshot.cn/ Accessed: 2025-02-05.

[5] Perplexity AI. 2024. Perplexity AI. https://www.perplexity.ai/ Accessed: 2025-02-05.

[6] Tiangong AI. 2024. Tiangong AI. http://tiangong.cn/ Accessed: 2025-02-05.

[7] Tongyi AI. 2024. Tongyi. https://tongyi.ai/ Accessed: 2025-02-05.

[8] Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. 2024. A Survey on Data Selection for Language Models. Transactions on Machine Learning Research (2024). https://openreview.net/forum?id=XfHWcNTSHp Survey Certification.

[9] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In The Twelfth International Conference on Learning Representations.

[10] Orlando Ayala and Patrice Bechard. 2024. Reducing hallucination in structured outputs via Retrieval-Augmented Generation. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track), Yi Yang, Aida Davani, Avi Sil, and Anoop Kumar (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 228–238. https://doi.org/10.18653/v1/2024.naacl-industry.19

[11] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. arXiv preprint arXiv:1611.09268 (2016).

[12] Alec Berntson. 2023. Azure AI Search: Outperforming Vector Search with Hybrid Retrieval and Reranking. https://techcommunity.microsoft.com/blog/azure-ai-services-blog/azure-ai-search-outperforming-vector-search-with-hybrid-retrieval-and-reranking/3929167 Accessed: 2025-02-01.

[13] Sergey Brin and Lawrence Page. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Networks 30 (1998), 107–117.

[14] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. Advances in neural information processing systems 33 (2020), 1877–1901.

[15] Harrison Chase. 2022. LangChain. https://github.com/langchain-ai/langchain

[16] ChatGLM. 2024. ChatGLM. https://chatglm.cn/ Accessed: 2025-02-05.

[17] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. arXiv:2402.03216 [cs.CL]

[18] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. Advances in neural information processing systems 30 (2017).

[19] Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. 719–729.

[20] Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: Open-source legal large language model with integrated external knowledge bases. CoRR (2023).

[21] Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018).

[22] Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-Verification Reduces Hallucination in Large Language Models. In Findings of the Association for Computational Linguistics: ACL 2024, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 3563–3578.

[23] Xinya Du and Heng Ji. 2022. Retrieval-Augmented Generative Question Answering for Event Argument Extraction. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 4649–4666. https://doi.org/10.18653/v1/2022.emnlp-main.307

[24] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise Zero-Shot Dense Retrieval without Relevance Labels. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 1762–1777.

[25] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. arXiv preprint arXiv:2303.14524 (2023).

[26] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997 [cs.CL]

[27] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. In The Thirty-eighth Annual Conference on Neural Information Processing Systems.

[28] Qisheng Hu, Geonsik Moon, and Hwee Tou Ng. 2024. From Moments to Milestones: Incremental Timeline Summarization Leveraging Large Language Models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 7232–7246. https://doi.org/10.18653/v1/2024.acl-long.390

[29] IBM. 2024. Metadata Enrichment: Highly Scalable Data Classification and Data Discovery. https://www.ibm.com/think/insights/metadata-enrichment-highly-scalable-data-classification-and-data-discovery Accessed: 2025-01-19.

[30] Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. arXiv preprint arXiv:2305.03653 (2023).

[31] Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2024. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. Advances in Neural Information Processing Systems 36 (2024).

[32] Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. LongLLMLingua: Accelerating and Enhancing LLMs in Long Context Scenarios via Prompt Compression. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 1658–1677. https://doi.org/10.18653/v1/2024.acl-long.91

[33] Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active Retrieval Augmented Generation. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 7969–7992. https://doi.org/10.18653/v1/2023.emnlp-main.495

[34] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. arXiv preprint arXiv:2503.09516 (2025).

[35] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. arXiv preprint arXiv:2001.08361 (2020).

[36] Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. 2023. Tree of Clarifications: Answering Ambiguous Questions with Retrieval-Augmented Large Language Models. In The 2023 Conference on Empirical Methods in Natural Language Processing.

[37] H. W. Kuhn. 1955. The Hungarian method for the assignment problem. Naval Research Logistics Quarterly 2, 1-2 (1955), 83–97. https://doi.org/10.1002/nav.3800020109 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.3800020109

[38] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems 33 (2020), 9459–9474.

[39] Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023. Making Large Language Models A Better Foundation For Dense Retrieval. arXiv:2312.15503 [cs.CL]

[40] Xinze Li, Zhenghao Liu, Chenyan Xiong, Shi Yu, Yu Gu, Zhiyuan Liu, and Ge Yu. 2023. Structure-Aware Language Model Pretraining Improves Dense Retrieval on Structured Data. In Findings of the Association for Computational Linguistics: ACL 2023, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 11560–11574. https://doi.org/10.18653/v1/2023.findings-acl.734

[41] Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2024. Chain-of-Knowledge: Grounding Large Language Models via Dynamic Knowledge Adapting over Heterogeneous Sources. In The Twelfth International Conference on Learning Representations.

[42] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the Middle: How Language Models Use Long Contexts. Transactions of the Association for Computational Linguistics 12 (2024), 157–173. https://doi.org/10.1162/tacl_a_00638

[43] Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 364 (2019).

[44] Ziyang Luo, Can Xu, Pu Zhao, Xiubo Geng, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. Augmented large language models with parametric

knowledge guiding. arXiv preprint arXiv:2305.04757 (2023).
[45] Xinbei Ma, Yeyun Gong, Pengcheng He, hai zhao, and Nan Duan. 2023. Query Rewriting in Retrieval-Augmented Large Language Models. In The 2023 Conference on Empirical Methods in Natural Language Processing. https://openreview.net/forum?id=gXq1cwkUZc
[46] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query Rewriting in Retrieval-Augmented Large Language Models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 5303–5315.
[47] Yubo Ma, Yixin Cao, Yong Ching Hong, and Aixin Sun. 2023. Large Language Model Is Not a Good Few-shot Information Extractor, but a Good Reranker for Hard Samples!. In The 2023 Conference on Empirical Methods in Natural Language Processing.
[48] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to Information Retrieval. Cambridge University Press.
[49] Richard E. Mayer. 2014. Cognitive Theory of Multimedia Learning. In The Cambridge Handbook of Multimedia Learning, Richard E. Mayer (Ed.). Cambridge University Press, Cambridge, 43–71.
[50] Metaso. 2024. Metaso. https://metaso.cn/ Accessed: 2025-02-05.
[51] Microsoft Azure Architecture Center. 2024. Developing a RAG Solution - Chunk Enrichment Phase. https://learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/rag/rag-enrichment-phase Accessed: 2025-01-19.
[52] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. arXiv preprint arXiv:2112.09332 (2021).
[53] Shiyu Ni, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2024. When Do LLMs Need Retrieval Augmentation? Mitigating LLMs' Overconfidence Helps Retrieval Augmentation. In Findings of the Association for Computational Linguistics: ACL 2024, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 11375–11388. https://doi.org/10.18653/v1/2024.findings-acl.675
[54] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018).
[55] OpenAI. 2024. ChatGPT. https://chatgpt.com/ Accessed: 2025-02-05.
[56] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. Advances in neural information processing systems 35 (2022), 27730–27744.
[57] Wenjun Peng, Guiyang Li, Yue Jiang, Zilong Wang, Dan Ou, Xiaoyi Zeng, Derong Xu, Tong Xu, and Enhong Chen. 2024. Large language model based long-tail query rewriting in taobao search. In Companion Proceedings of the ACM on Web Conference 2024. 20–28.
[58] Luis P. Prieto, Kshitij Sharma, Łukasz Kidzinski, María Jesús Rodríguez-Triana, and Pierre Dillenbourg. 2018. Multimodal Teaching Analytics: Automated Extraction of Orchestration Graphs from Wearable Sensor Data. Journal of Computer Assisted Learning 34, 2 (April 2018), 193–203. https://doi.org/10.1111/jcal.12232
[59] Alec Radford and Karthik Narasimhan. 2018. Improving Language Understanding by Generative Pre-Training. https://api.semanticscholar.org/CorpusID:49313245
[60] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019).
[61] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Journal of Machine Learning Research 21, 140 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html
[62] Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing Retrieval-Augmented Large Language Models with Iterative Retrieval-Generation Synergy. In Findings of the Association for Computational Linguistics: EMNLP 2023, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 9248–9274. https://doi.org/10.18653/v1/2023.findings-emnlp.620
[63] Statista. 2023. https://www.statista.com/statistics/1377993/us-adults-ai-powered-search-engines-usage-choice/ Accessed: 2025-01-21.
[64] Winnie Street, John Oliver Siy, Geoff Keeling, Adrien Baranes, Benjamin Barnett, Michael McKibben, Tatenda Kanyere, Alison Lentz, Robin IM Dunbar, et al. 2024. LLMs achieve adult human performance on higher-order theory of mind tasks. arXiv preprint arXiv:2405.18870 (2024).
[65] Hao Sun, Zhexin Zhang, Jiawen Deng, Jiale Cheng, and Minlie Huang. 2023. Safety assessment of chinese large language models. arXiv preprint arXiv:2304.10436 (2023).
[66] John Sweller, Paul Ayres, and Slava Kalyuga. 2020. Cognitive load theory and educational technology. Educational Technology Research and Development 68, 1 (2020), 1–16. https://doi.org/10.1007/s11423-019-09701-3
[67] Robert Endre Tarjan and Anthony E Trojanowski. 1977. Finding a maximum independent set. SIAM J. Comput. 6, 3 (1977), 537–546.
[68] Ravi Theja. 2023. Evaluating the Ideal Chunk Size for a RAG System using LlamaIndex. https://www.llamaindex.ai/blog/evaluating-the-ideal-chunk-size-for-a-rag-system-using-llamaindex-6207e5d3fec5 Accessed: 2025-01-19.
[69] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 10014–10037. https://doi.org/10.18653/v1/2023.acl-long.557
[70] A Vaswani. 2017. Attention is all you need. Advances in Neural Information Processing Systems (2017).
[71] Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query Expansion with Large Language Models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 9414–9423. https://doi.org/10.18653/v1/2023.emnlp-main.585
[72] Xintao Wang, Qianwen Yang, Yongting Qiu, Jiaqing Liang, Qianyu He, Zhouhong Gu, Yanghua Xiao, and Wei Wang. 2023. Knowledgpt: Enhancing large language models with retrieval and storage access on knowledge bases. arXiv preprint arXiv:2308.11761 (2023).
[73] Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. 2023. Do-Not-Answer: A Dataset for Evaluating Safeguards in LLMs. arXiv preprint arXiv:2308.13387.
[74] Weiqi Wu, Shen Huang, Yong Jiang, Pengjun Xie, Fei Huang, and Hai Zhao. 2025. Unfolding the Headline: Iterative Self-Questioning for News Retrieval and Timeline Summarization. arXiv preprint arXiv:2501.00888 (2025).
[75] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. arXiv:2309.07597 [cs.CL]
[76] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. arXiv preprint arXiv:2412.15115 (2024).
[77] Haoyan Yang, Zhitao Li, Yong Zhang, Jianzong Wang, Ning Cheng, Ming Li, and Jing Xiao. 2023. PRCA: Fitting Black-Box Large Language Models for Retrieval Question Answering via Pluggable Reward-Driven Contextual Adapter. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 5364–5375. https://doi.org/10.18653/v1/2023.emnlp-main.326
[78] Zhilin Yang. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. arXiv preprint arXiv:1906.08237 (2019).
[79] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, Brussels, Belgium, 2369–2380. https://doi.org/10.18653/v1/D18-1259
[80] Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making Retrieval-Augmented Language Models Robust to Irrelevant Context. In The Twelfth International Conference on Learning Representations.
[81] Liangyu Zha, Junlin Zhou, Liyao Li, Rui Wang, Qingyi Huang, Saisai Yang, Jing Yuan, Changbao Su, Xiang Li, Aofeng Su, et al. 2023. Tablegpt: Towards unifying tables, nature language and commands into one gpt. arXiv preprint arXiv:2307.08674 (2023).
[82] Kun Zhang, Jiali Zeng, Fandong Meng, Yuanzhuo Wang, Shiqi Sun, Long Bai, Huawei Shen, and Jie Zhou. 2024. Tree-of-Reasoning Question Decomposition for Complex Question Answering with Large Language Models. Proceedings of the AAAI Conference on Artificial Intelligence 38, 17 (Mar. 2024), 19560–19568. https://doi.org/10.1609/aaai.v38i17.29928
[83] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. arXiv preprint arXiv:2309.01219 (2023).
[84] Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. 2024. Take a Step Back: Evoking Reasoning via Abstraction in Large Language Models. In The Twelfth International Conference on Learning Representations.
[85] Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. arXiv preprint arXiv:2504.03160 (2025).
[86] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. In The Eleventh International Conference on Learning Representations. https://openreview.net/forum?id=WZH7099tgfM

[87] Junyi Zhu, Shuochen Liu, Yu Yu, Bo Tang, Yibo Yan, Zhiyu Li, Feiyu Xiong, Tong Xu, and Matthew B. Blaschko. 2024. FastMem: Fast Memorization of Prompt Improves Context Awareness of Large Language Models. In Findings of the Association for Computational Linguistics: EMNLP 2024, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 11740–11758. https://doi.org/10.18653/v1/2024.findings-emnlp.687

[88] Shengyao Zhuang, Bing Liu, Bevan Koopman, and Guido Zuccon. 2023. Open-source Large Language Models are Strong Zero-shot Query Likelihood Models for Document Ranking. In Findings of the Association for Computational Linguistics: EMNLP 2023, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 8807–8817.

Supplementary Materials

A MULTI-FACETED EVALUATION CRITERIA

The detailed definitions of the multi-faceted evaluation criteria are provided in Tab. 11. The original text is in Chinese, and we translate it into English to align with the language of this paper.

B INSTRUCTION PROMPT

B.1 Graph-based Query Decomposition

Prompt for the graph-based query decomposition is provided in Tab. 12.

B.2 Answer Generation

Prompt for answer generation is provided in Tab. 13.

B.3 LLM Evaluation

Tab. 14 presents the prompt used to instruct the LLM to evaluate the generated text based on multi-faceted criteria (see Tab. 11). To reduce task complexity and enhance evaluation quality, we assess one facet per evaluation and fill the metric title and definition accordingly.

B.4 Built-In Citation

Prompt for entity extraction is provided in Tab. 15. Prompt for citation identification is provided in Tab. 16.

C RETRIEVAL DOCUMENTS FILTERING RULES

HTML Content Filtering. Non-content HTML tags, such as <script> and <style>, are removed using the lxml library to parse the original HTML into a DOM tree. Using XPath selectors, we retain only essential content tags like , <p>, and <article>, ensuring compatibility with irregular HTML structures.

Text Processing. : Extracted text blocks are separated by spaces or line breaks to improve readability. Redundant whitespace and excessive line breaks are removed while preserving paragraph structure. Distracting elements like "Read More," "Click to Continue," or inline emojis are filtered out. Special characters, stopwords (e.g., from publicly available resources like Stopwords JSON[1]), emoji patterns (via regular expressions targeting Unicode Emoji ranges), and irrelevant newline characters are eliminated.

Sensitive Information Filtering. Personal identifiers, such as phone numbers, email addresses, and platform-specific markers are detected and removed.

Text Normalization. Punctuation is standardized to half-width characters, and numbers in various formats are converted to standardized half-width Arabic numerals.

Table 11: Multi-faceted evaluation criteria.

```
(1) Conciseness:
    - The response should directly address the user's
        question.
    - Avoid irrelevant content, unnecessary information,
        or roundabout explanations.
    - Deduct 1 point for each irrelevant statement.

(2) Numerical Precision:
    - If a question requires a specific number, avoid
        vague terms like "several" or "many times."
    - Responses should be precise and specific.
    - Deduct 1 point for each ambiguous statement.

(3) Relevance:
    - If the question specifies constraints (e.g., time,
        location, person, event),
      the answer must adhere to them.
    - Deduct 1 point for each instance of misalignment
        with the question's constraints.

(4) Factuality:
    - The information must be factually correct,
        especially for numerical or factual questions.
    - Deduct 1 point for each incorrect numerical or
        factual statement.

(5) Timeliness:
    - For ongoing news or urgent reports, ensure
        information reflects the latest updates.
    - The current date is {to be filled}.
    - If the question is not time-sensitive, no points
        are deducted.
    - For time-sensitive questions, deduct points
        proportionally based on outdatedness.

(6) Comprehensiveness:
    - The response should comprehensively cover all
        aspects of the user's inquiry.
    - The user should not need further search to grasp
        the full context.
    - Deduct 1 point for each missing essential element.

(7) Clarity:
    - The response should be easy to understand, well-
        structured, and formatted logically.
    - Example: Chronological events should be presented
        in chronological order.
    - Deduct 1 point for unclear or disorganized
        presentation.

(8) Coherence:
    - The response should be logically consistent, with
        smooth transitions between sentences.
    - Deduct 1 point for each instance of incoherent or
        disjointed phrasing.

(9) Insightfulness:
    - The response should provide insightful or unique
        perspectives.
    - Base score: 6 points.
    - Award 0.5-1 additional points for each innovative
        idea or expression.
```

[1]https://github.com/6/stopwords-json/blob/master/dist/ca.json

Table 12: Prompt for graph-based query decomposition.

```
Please analyze the following query and return the
explanation in dictionary format.

Response format:
{'is_complex': True/False, 'sub_queries': [], 'parent_child': []}

Analysis Steps and Principles:

1. **Classify the nature of the query**
   - The query can be classified into one of two types:
     (a) A "complex query" that consists of multiple sub-queries.
     (b) A "simple query" that can be directly answered.
   - If the query is classified as "complex," set 'is_complex' to **True**.
   - If the query is "simple," set 'is_complex' to **False**, and leave 'sub_queries' and 'parent_child' as empty lists.

2. **Decomposing a Complex Query**
   - If the query is classified as "complex," break it down into **sub-queries** and store them in the 'sub_queries'
       list.
   - Decomposition principles:
     1) If a query contains multiple **target entities**, split it into multiple sub-queries.
        - Example: *"What are the latest social news and weather news in Shanghai?"*
          - Target entities: *"social news"*, *"weather news"*.
          - Split into: *"What are the latest social news in Shanghai?"* and *"What are the latest weather news in
              Shanghai?"*.
     2) Each sub-query should be **indivisible** and should not require further decomposition.
     3) No duplicate sub-queries.
     4) When referring to **names of people, places, or organizations**, ensure full and precise descriptions.
        - Example: *"What is the area and population of New Jersey, USA?"*
          - Correct split: *"What is the area of New Jersey, USA?"* and *"What is the population of New Jersey, USA?"*.
          - Incorrect split: *"What is the area of New Jersey?"* and *"What is the population of New Jersey?"*.
     5) The total number of sub-queries **should not exceed 6**.

3. **Analyzing Dependencies Between Sub-Queries**
   - If the query is complex, analyze the **dependency relationships** between sub-queries and store them in '
       parent_child'.
   - Example:
     - *"What natural disasters occurred in Indonesia in April?"*
     - *"How long did this natural disaster last?"*
     - The second question **depends** on the first; thus, the first is the *parent*, and the second is the *child*:
       ```json
       {"parent": "What natural disasters occurred in Indonesia in April?",
        "child": "How long did this natural disaster last?"}
       ```
   - Dependency principles:
     1) If sub-queries are **independent**, 'parent_child' remains an empty list.
     2) If the **child question cannot be answered without the parent**, it is a dependent relationship.
        - Example: "What is the latest iPhone model" is the parent node of "What are the specifications of the latest
            iPhone?"
        - The first question must be answered before the second.
     3) Every possible pair of sub-queries should be evaluated for dependency.
        - A query can be both a *parent* and a *child* in different relationships.

Example:
{Few-Shot Examples}

Query: {Query}
Response: \n
```

D FINE-TUNING

D.1 User Intent Understanding

To promote responsible AI behavior and constrain the scope of queries, we fine-tune the model for the query rejection. We define 11 categories of queries that warrant refusal, including: (1) illegal content, (2) ethical violations, (3) privacy breaches, (4) harmful intent, (5) professional consultations, (6) human-AI interactions, (7) misinformation, (8) technical inquiries, (9) academic requests, (10) planning and consulting inquiries, and (11) creative content generation.

To construct a training dataset, we collect a set of seed queries based on open-source datasets: Do-Not-Answer [73], BeaverTails [31] and Safety-Prompts [65]. Additionally, we generate synthetic queries to compensate the imbalance number for each class in the collected dataset. Then we instruct multiple LLMs by given the definition of the 11 categories to classify and output in JSON format as follows:

```
{
“Refusal”: “Yes/No”,
“Category”: “illegal content/ethical violations/. . . ”
}
```

Table 13: Prompt for answer generation.

```
You are an AI assistant named Xinyu, developed by the
Shanghai Algorithm Innovation Research Institute. You are
performing an encyclopedia Q\&A task. Please generate an
answer based on the provided reference materials and
related Q\&A content.

Question: {Sub-Query}

Related Q\&A:
{Ancestor Node 1: Sub-Query}
{Ancestor Node 1: Answer}
{Ancestor Node 2: Sub-Query}
{Ancestor Node 2: Answer}
...

Reference materials:
{Retrieved Passage 1}
{Retrieved Passage 2}
...

When generating your answer, follow these guidelines:

[Structural Requirements]:
To ensure clarity and organization, you may use one or
more of the following structured formats:
 - **Introduction-Body-Summary**: Introduce the topic,
     elaborate, and summarize key points.
 - **Paragraphs by Subquestion**: Address each
     subquestion in a separate paragraph.
 - **Cause and Effect**: Explain the causes and
     consequences of an event.
 - **Comparison and Contrast**: Describe and compare two
     or more concepts.
 - **Chronological Order**: Describe events or steps in
     order of occurrence.
 - **Problem-Solution**: Introduce a problem and explain
     solutions or strategies.
 - **Pros and Cons**: List the positive and negative
     aspects of a decision or choice.
 - **Definition and Examples**: Provide a definition and
     illustrate it with examples.
 - **Logical Reasoning**: Derive conclusions based on
     assumptions or premises.
 - **List Structure**: Enumerate facts or features for
     easy readability.
 - **Categorization**: Introduce a concept, group it by
     categories, and explain in detail.
 - **Theme and Variations**: Explore a core theme and its
      variations.
 - **Case Study**: Explain a theory or concept through
     specific cases.
 - **Hierarchical Structure**: Arrange information by
     importance or sequence.
 - **Issue and Counterarguments**: Present an issue with
     supporting and opposing views.

[Language Requirements]:
(1) Use concise and clear language.
(2) Ensure that the answer's structure enhances clarity
     and readability.
(3) The response must directly and accurately address the
     question, avoiding irrelevant content.
(4) When citing reference materials, ignore template
     formatting or improper phrasing.
(5) If detailed elaboration is required, output the
     answer in a structured **Markdown** format.

Your Answer: \n
```

Decisions are aggregated using a majority voting mechanism to establish a consensus. Finally, human experts review the results to correct misclassification.

Table 14: Prompt for multi-faceted evaluation.

```
Assume you are an article quality inspector. Please
evaluate the response based on {Metric Title}.
I will provide the user's question and the final response
The maximum score is 10 points, and the scoring rules are
as follows:

{Metric Definition}

Please strictly follow the scoring rules. Example output
format:

'{
    "Issues Identified": "X",
    "Calculation Process": "10-1.0-1.0-1.0 = 7.0",
    "Score": 7
}'

{Few-Shot Examples}

Your final score: \n"
```

Table 15: Prompt for entity extraction.

```
Read the given sentence and extract the contained
information about time, location, persons, and job titles
     .

Your extraction result should be returned in JSON format,
with each field name restricted to one of the following:
["Time", "Location", "Persons", "Job Titles"]

If there are multiple pieces of information of the same
type in the sentence, the corresponding category's value
should be represented as an array.

Below are some examples:

{Few-Shot Examples}

{Sentence}

Extraction result: \n
```

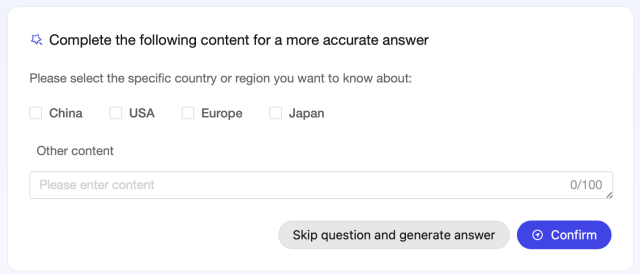

Figure 5: Xinyu's Interface for query disambiguation.

For queries classified as non-refusal, we further prepare a dataset for query disambiguation. We instruct multiple LLMs to analyze whether a query requires further clarification to generate an appropriate response, outputting the results in the following JSON format:

```
{
  "Requires additional input": "Yes/No",
```

Table 16: Prompt for citation identification.

```
You are a journalist skilled in analyzing the correlation
between document information. I will provide you with a
sentence excerpted from a news article, along with
several reference documents used in writing this article.
Your task is to determine which reference document the
excerpted sentence most likely originates from.

The excerpted sentence is:
{Sentence}

The key information contained in this sentence is:
Time: {Time}
Location: {Location}
Person: {Person}
Job Title: {Job Title}
Numbers: {Numbers}

The reference documents used for writing this article and
their respective key information are as follows:
[1] {Retrieved Document}
[2] {Retrieved Document}
[3] ...

When making your determination, ensure that the selected
reference document matches as much key information from
the excerpted sentence as possible. The higher the degree
of key information overlap, the more likely the reference
document is the source of the excerpted sentence.

Your response should contain only a one- or two-digit
number representing the corresponding reference document
number, such as "[2]", "[9]", or "[13]". If you believe
that none of the reference documents are relevant to the
given sentence, return "-1".

The most likely source document number is: \n
```

```
  "Additional options": {
    "Prompt description": "Please select...",
    "Choices": ["xx", "xx", ...]
  }
}
```

Additional options are prepared to present clarifying options for the user as shown in Fig. 5. Again, we use a majority voting mechanism to determine whether additional input is required. This process results in 7K data points. Human experts then review the results, refine the choices, and select high-quality samples. Ultimately, we construct a dataset of 5K data points, where 1/4 of the queries require additional input.

Two models are fine-tuned for the query rejection and query disambiguation tasks using their respective datasets, following Eq. (1).

D.2 Reranker Model

We construct a dataset comprising question-answer pairs using recent real-world data and public datasets such as MS MARCO [11]. To generate hard negatives for fine-tuning, we apply various chunking strategies to create multiple candidate samples resembling the positive examples. These candidates are then ranked based on a base reranker model, selecting the top-300 samples. Next, we leverage multiple LLMs to assess whether each generated sample can answer the corresponding question. If the majority vote is negative, the sample is designated as a hard negative. In total, we generate 56K pairs. Finally, human experts review the results and curate a high-quality subset, retaining 13K pairs for fine-tuning.

Table 17: Comparison of answers generated with or without GQD.

```
Query: "Analyze how 'Blue Moon' and 'Super Moon' have
     evolved in cultural and folk traditions throughout
     history."

Answer with GQD:
In ancient times, a 'Blue Moon' was often seen as an
ominous sign linked to disasters, reflecting early
superstition. Over centuries, it evolved from referring
to the fourth full moon in one season to the second full
moon in a calendar month-highlighting changes in
calendars and beliefs. Likewise, early Chinese traditions
revered a Super Moon as a symbol of cosmic harmony,
whereas modern science explains it simply as a near-
perigee full moon. This trajectory captures a shift from
mystical interpretation to scientific understanding in
various cultures...

Answer without GQD:
Historically, 'Blue Moon' and 'Super Moon' did not draw
much cultural attention; ancient societies were more
generally focused on lunar cycles. These terms gained
modern popularity mainly due to social media, prompting
renewed public interest.
```

D.3 Built-In Citation

As this module requires high efficiency, we fine-tune two SLMs (Qwen2.5-3B), using its larger counterpart, Qwen2.5-72B. We collect a set of passages and use Qwen2.5-72B to extract the entities from each sentence based on the prompt provided in systems Supplementary Material B.4. If any entities can be extracted, we retain the (passage, sentence, entities) triplet as part of the dataset. This process yields 33K data points. Human experts then review the data to correct errors and remove low-quality samples, ultimately retaining 26K data points. We fine-tune the entity extraction SLM using sentences paired with their corresponding entities. Additionally, we fine-tune another SLM to retrieve the relevant passage from a given set based on the extracted entities, using the prompt provided in Supplementary Material B.4.

E ADDITIONAL ABLATION RESULTS

Multi-Faceted Evaluation by LLM. Tab. 18 presents the results of a multi-faceted evaluation conducted by GPT-4O. While the absolute values differ from those obtained through human evaluation (see Tab. 1), the rankings remain similar, demonstrating a strong correlation (see Tab. 2). These results confirm that LLM-based evaluation is also indicative of performance.

Table 18: Multi-faceted comparison of different approaches based on GPT-4O. Higher value indicates better performance, 10 is the maximum.

Model	Conciseness	Numerical Precision	Relevance	Factuality	Timeliness	Comprehensiveness	Clarity	Coherence	Insightfulness	Average
Perplexity AI	**9.913**	**9.607**	**9.740**	9.727	8.120	8.280	9.887	**9.853**	6.613	9.082
Tiangong AI [6]	9.819	9.188	9.570	9.738	7.758	7.517	9.839	9.799	6.161	8.821
Ernie Bot [2]	9.814	9.152	9.556	9.648	8.062	7.924	9.745	9.814	6.552	8.918
KIMI [4]	9.695	9.359	9.576	9.675	8.059	8.305	9.686	9.720	6.432	8.945
Metaso [50]	9.781	8.932	9.493	9.596	7.589	6.842	9.712	9.589	5.801	8.593
ChatGLM [16]	9.733	9.274	9.568	**9.745**	7.986	7.911	9.863	9.808	6.603	8.943
Baichuan [1]	9.433	9.053	9.307	9.403	7.813	7.832	9.373	9.200	6.640	8.673
Tongyi [7]	9.747	8.900	9.313	9.527	7.700	7.940	9.827	9.740	6.493	8.799
Xinyu (Ours)	9.880	9.547	9.547	9.731	**8.300**	**8.533**	**9.900**	9.747	**7.107**	**9.144**